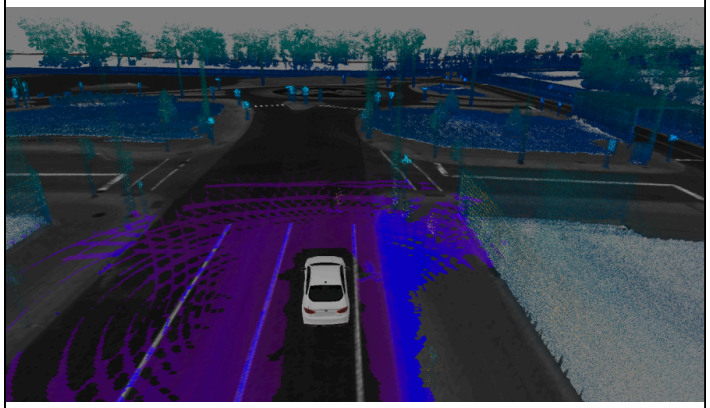
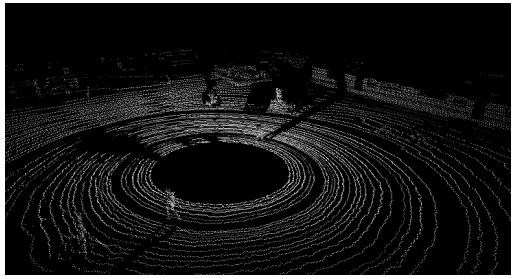
ML- Computer Vision

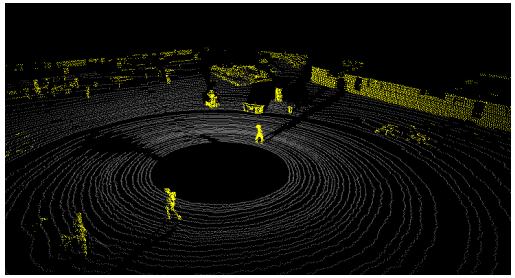
Shameless Plug



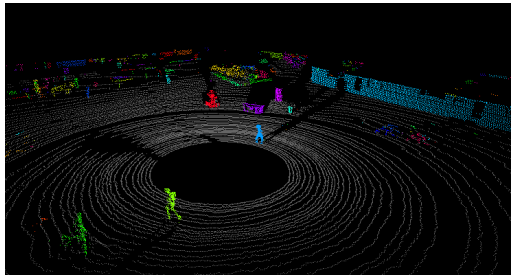
- Summer Opportunity – Self Driving Cars
  - Machine Learning
  - Image Classification
  - Control
  - Crowdsourcing
  - Visualization/OpenGL
  - GPU Programming
  - Simulation
  - Planning



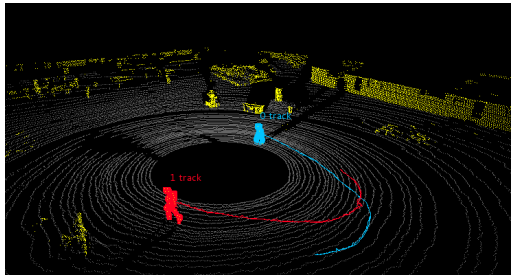
Courtesy P. Morton



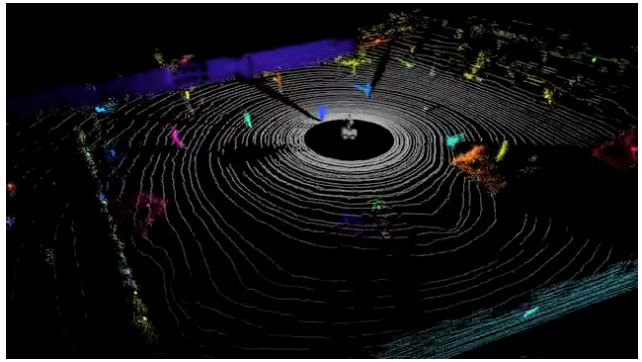
Courtesy P. Morton



Courtesy P. Morton



Courtesy P. Morton



- $5M University Project
  - 15-20 students
  - Possibility for corporate internships
  - Possibility follow-on employment

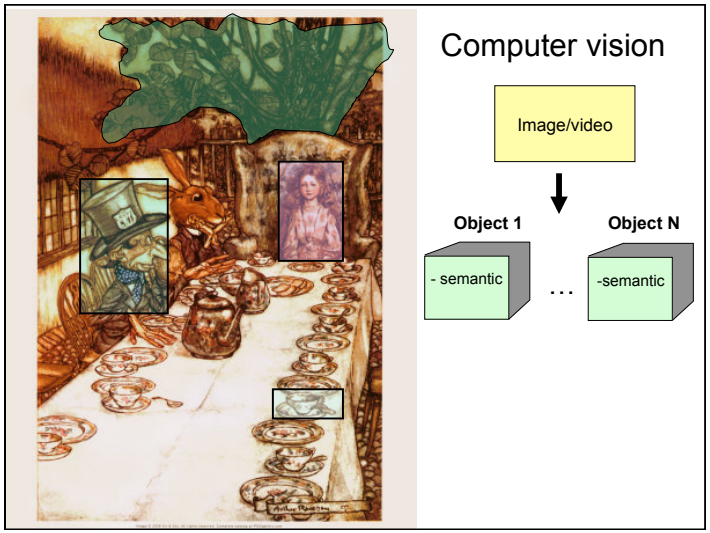
- If you are interested
  - Follow these instructions exactly
  - Write a 1 page document with:
    - Open with your availability for the summer and your current position undergrad/gradstudent 2nd year etc.
    - 1 paragraph on your background prior projects
    - 3 paragraphs on what you are interested in working on for the summer
  - Email it to both mattjr@umich.edu and ramv@umich.edu with the subject line "Ford NGV Summer Project"
  - This is our 0-th order test for your ability to deliver to spec!!!

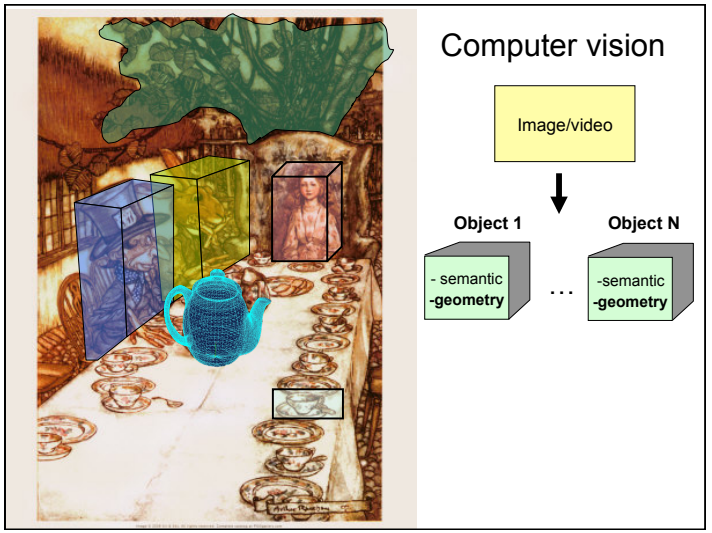
Vision

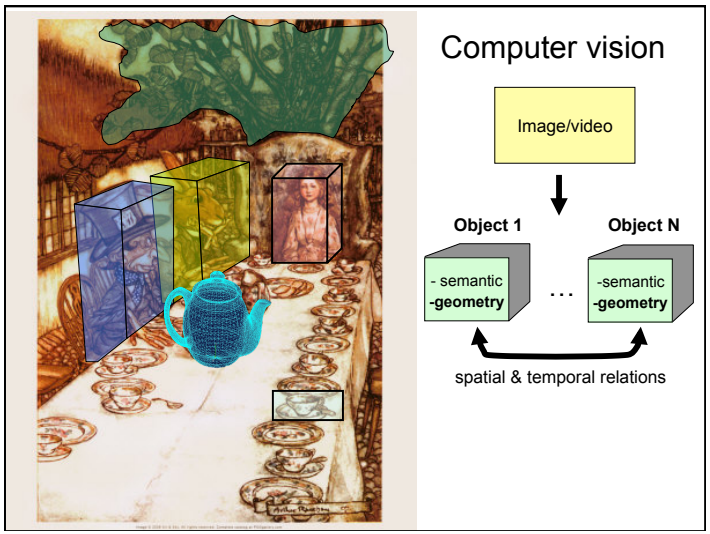
"There was a table set out under a tree in front of the house, and the March Hare and the Hatter were having tea at it."

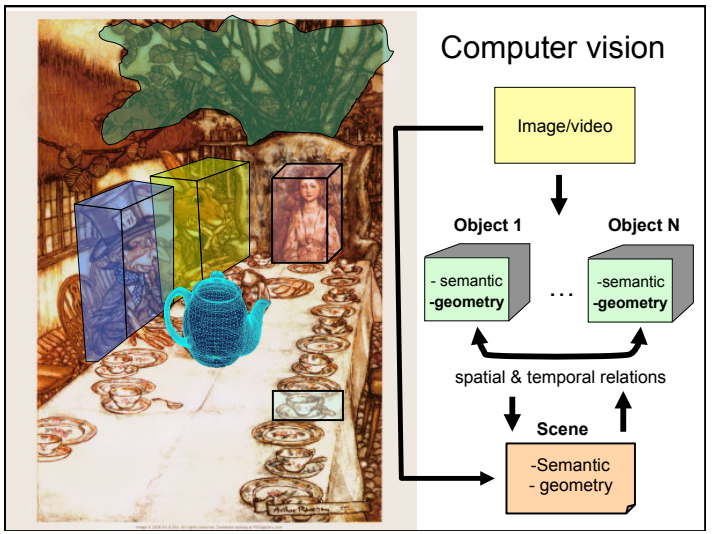
"The table was a large one, but the three were all crowded together at one corner of it …"

From "A Mad Tea-Party"
Alice's Adventures in Wonderland
by
Lewis Carroll


Computer vision
Image/video
Object 1
Object N
- semantic
...
-semantic


Computer vision
Image/video
Object 1
Object N
- semantic
-geometry
...
-semantic
-geometry


Computer vision
Image/video
Object 1
Object N
- semantic
-geometry
...
-semantic
-geometry
spatial & temporal relations


Computer vision
Image/video
Object 1
Object N
- semantic
-geometry
...
-semantic
-geometry
spatial & temporal relations
Scene
-Semantic
- geometry

Computer vision

Computer vision studies the tools and theories that enable the design of machines that can extract useful information from imagery data (images and videos) toward the goal of interpreting the world

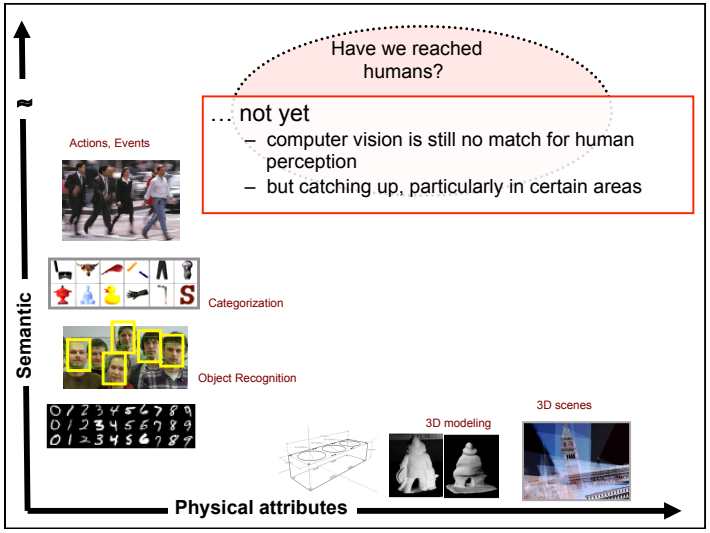
• Scene
• Objects
• People
• Actions

Sensing device

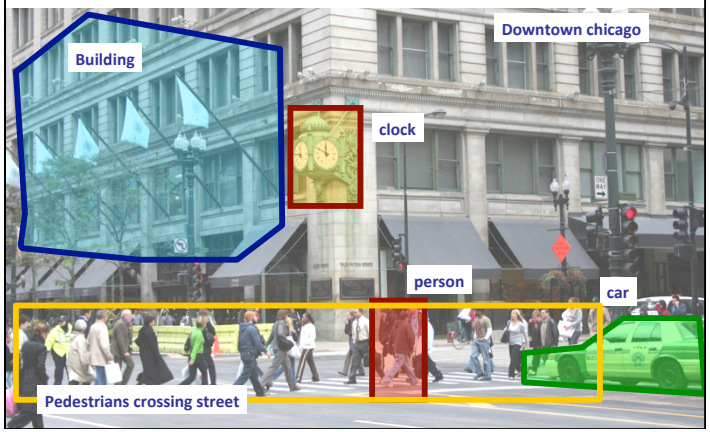
Computational device

•Extract information
• Interpretation

Information: visual cues, 3D structure, motion flows, etc…
Interpretation: recognize objects, scenes, actions, events



Object recognition and categorization



Challenges: viewpoint variation

Michelangelo 1475-1564

slide credit: Fei-Fei, Fergus & Torralba

Challenges: illumination

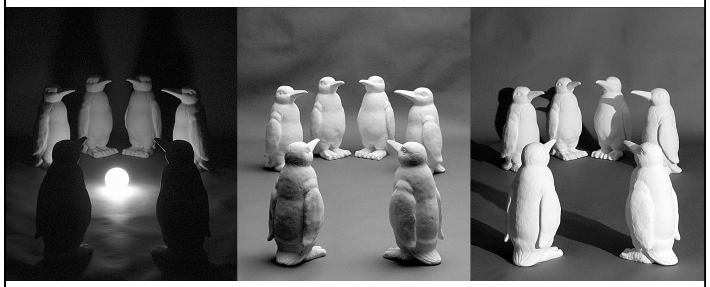


image credit: J. Koenderink

Challenges: scale

slide credit: Fei-Fei, Fergus & Torralba

Challenges: deformation

Challenges: occlusion

Magritte, 1957

slide credit: Fei-Fei, Fergus & Torralba

Challenges: background clutter

Kilmeny Niland. 1995

Challenges: object intra-class variation

slide credit: Fei-Fei, Fergus & Torralba

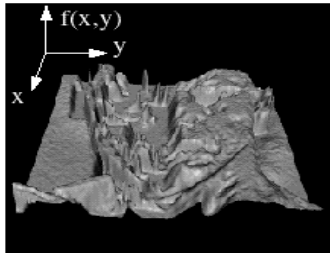
Images as functions

- We can think of an **image** as a function, *f*, from R^2 to R:
  - Defined over a rectangle, with a finite range:
    - f: $[a,b]$ x $[c,d]$ → $[0,255]$
  - $f(x, y)$ gives the **intensity** at position (x, y)
- A color image:

$$f(x,y) = \begin{bmatrix} r(x,y) \\ g(x,y) \\ b(x,y) \end{bmatrix}$$

Source: S. Seitz

Images as functions



Source: S. Seitz

Images as functions

- Images are usually **digital** (**discrete**):
  - **Sample** the 2D space on a regular grid
- The image can now be represented as a matrix of integer values

pixel

j

i

62	79	23	119	120	105	4	0
10	10	9	62	12	78	34	0
10	58	197	46	46	0	0	48
176	135	5	188	191	68	0	49
2	1	1	29	26	37	0	77
0	89	144	147	187	102	62	208
255	252	0	166	123	62	0	31
166	63	127	17	1	0	99	30

Source: S. Seitz

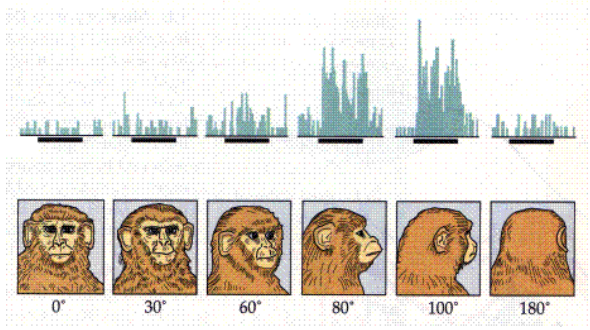
Face Recognition

- PCA (Eigen-faces)
- Boosting

Segments of this lectures are courtesy of Prof F. Li

Grandmother Cell

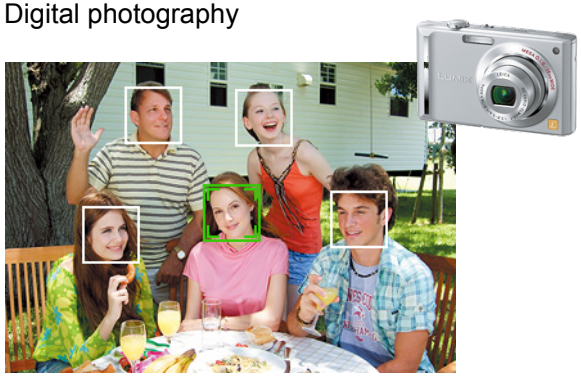
The **grandmother cell** is a hypothetical neuron that represents any complex and specific concept or object [Perrett et al 1989]



Courtesy of Johannes M. Zanker

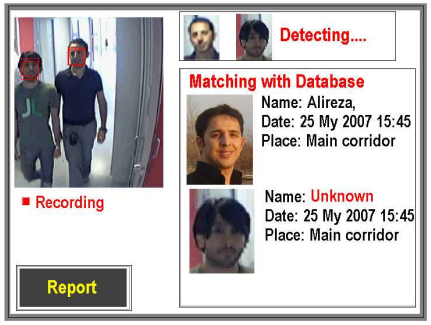
Face Recognition

- Digital photography



Face Recognition

- Digital photography
- Surveillance



Face Recognition

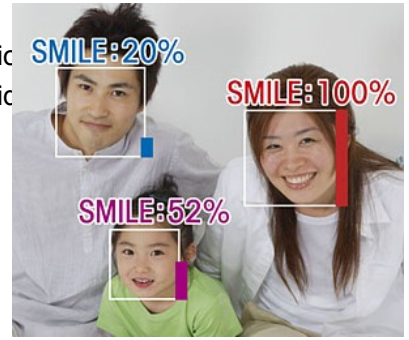
- Digital photography
- Surveillance
- Album organization

Face Recognition

- Digital photography
- Surveillance
- Album organization
- Person tracking/id.

Face Recognition

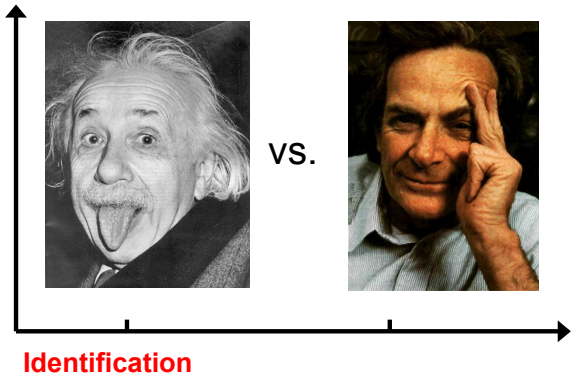
- Digital photography
- Surveillance
- Album organizatic
- Person tracking/ic
- Emotions and expressions



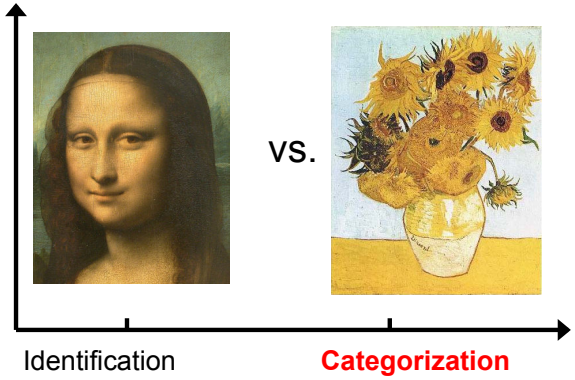
Face Recognition

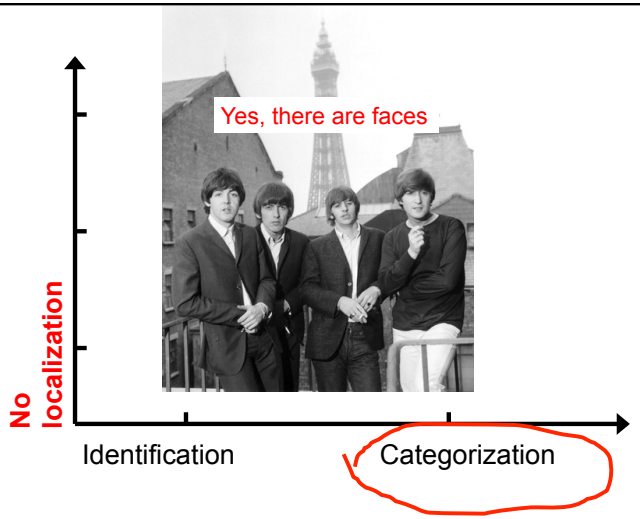
- Digital photography
- Surveillance
- Album organization
- Person tracking/id.
- Emotions and expressions
- Security/warfare
- Tele-conferencing
- Etc.

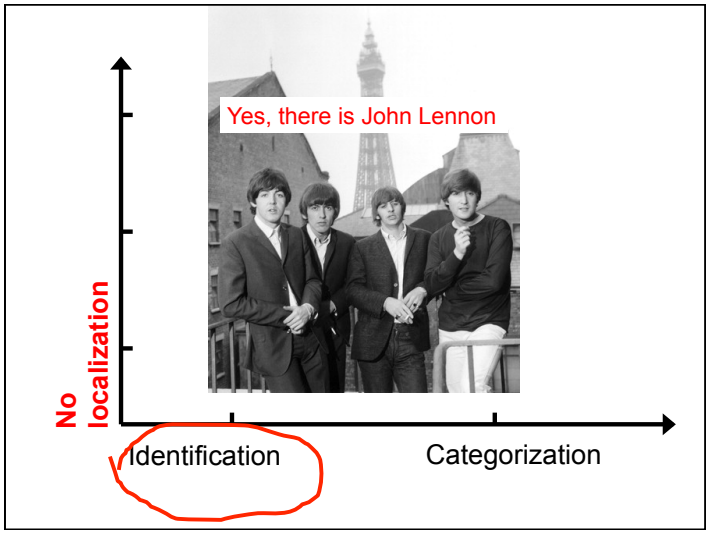
What's 'recognition'?

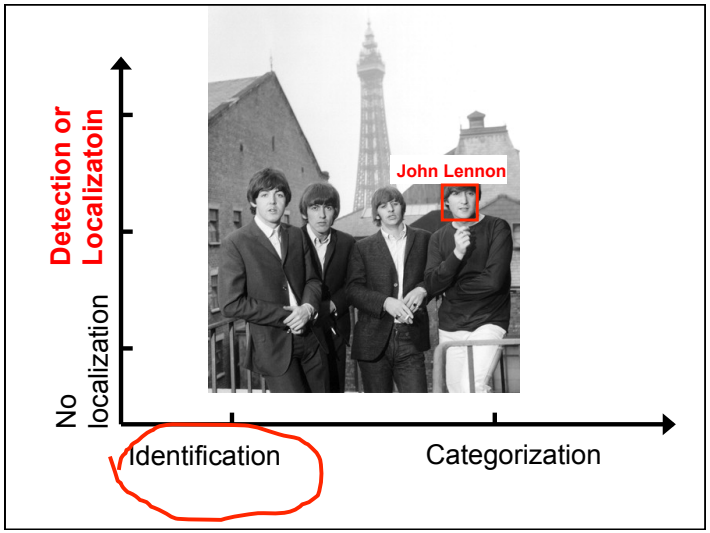


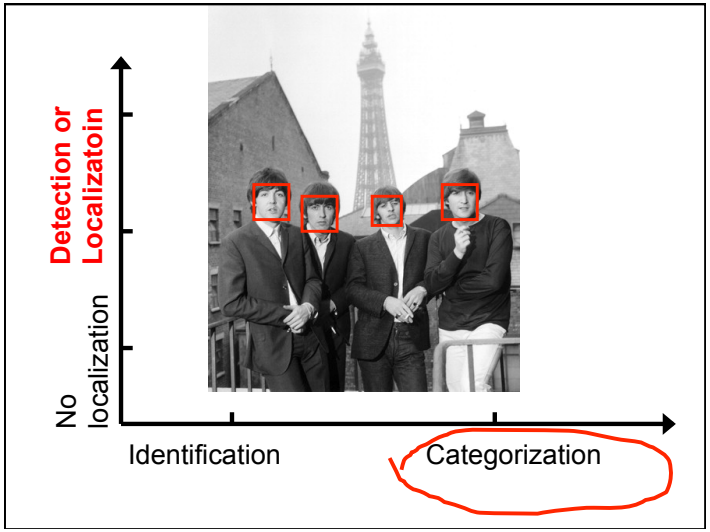
What's 'recognition'?

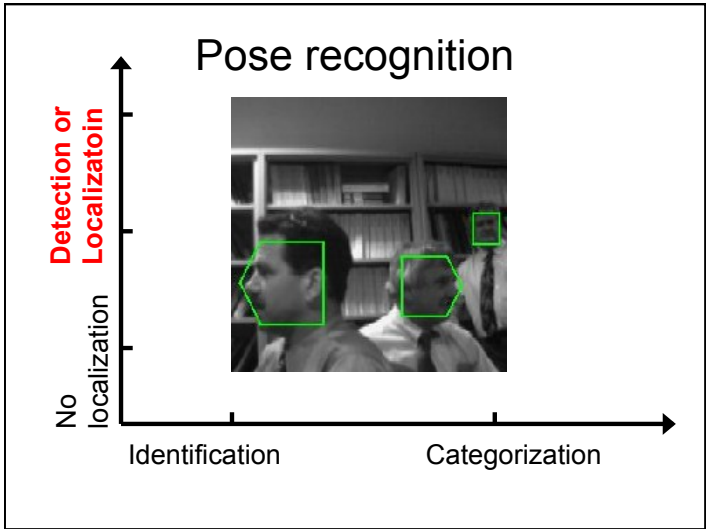




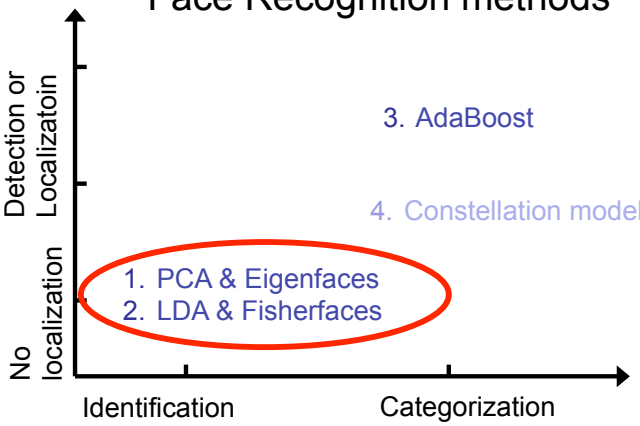

Yes, there is John Lennon
No localization
Identification
Categorization


John Lennon
Detection or Localizatoin
No localization
Identification
Categorization


Detection or Localizatoin
No localization
Identification
Categorization


Pose recognition
Detection or Localizatoin
No localization
Identification
Categorization

Face Recognition methods



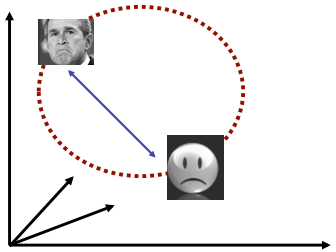
Eigenfaces and Fisherfaces

- Principle Component Analysis (PCA)
- Linear Discriminant Analysis (LDA)

References:

1. Turk and Penland, Eigenfaces for Recognition, 1991
2. Belhumeur, Hespanha and Kriegman, Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection

The Space of Faces



- An image is a point in a high dimensional space
  - An N x M image is a point in R^{NM}

[Thanks to Chuck Dyer, Steve Seitz, Nishino]

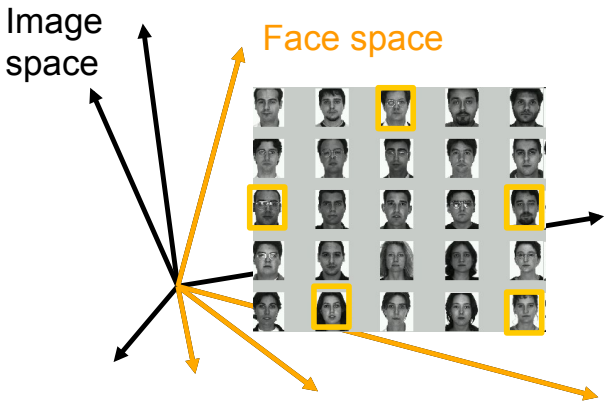
Key Idea

- Images in the possible set $\chi = \{\hat{x}\}$ are highly correlated.
- So, compress them to a low-dimensional subspace that captures key appearance characteristics of the visual DOFs.

- USE PCA for estimating the sub-space (dimensionality reduction)

•Compare two faces by projecting the images into the subspace and measuring the EUCLIDEAN distance between them.

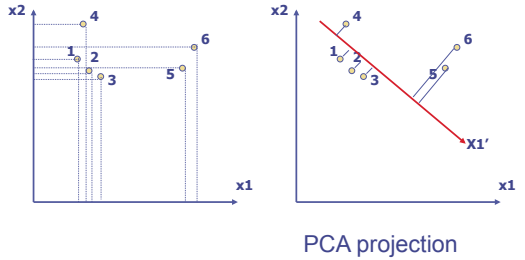
EIGENFACES: [Turk and Pentland 91]



• Computes n-dim subspace such that the projection of the data points onto the subspace has the largest variance among all n-dim subspaces.

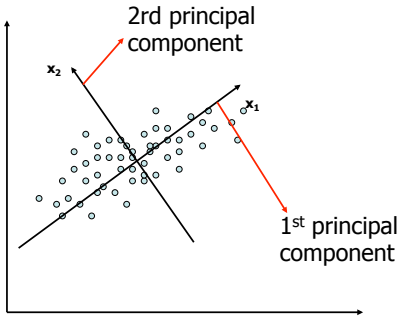
• Maximize the scatter of the training images in face space

USE PCA for estimating the sub-space



• Computes n-dim subspace such that the projection of the data points onto the subspace has the largest variance among all n-dim subspaces.

USE PCA for estimating the sub-space



PCA Mathematical Formulation

PCA = eigenvalue decomposition of a data covariance matrix

Define a transformation, W,

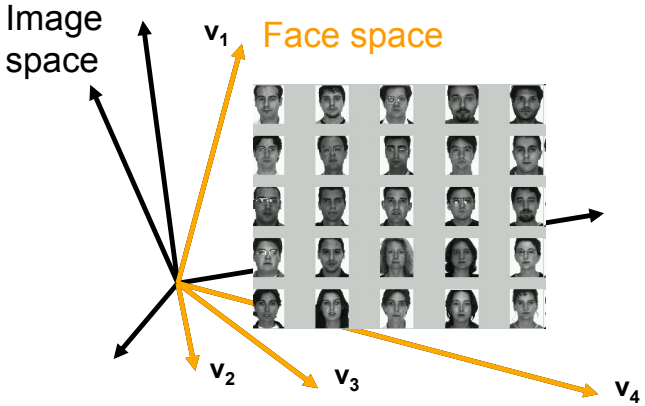
$$y_j = W^T x_j \qquad j = 1, 2 \ldots N$$

m-dimensional | Orthonor $W \in \mathbb{R}^{n \times m}$ | n-dimensional

$$S_T = \sum_{j=1}^{N} (x_j - \bar{x})(x_j - \bar{x})^T$$ = Data Scatter matrix

$$\tilde{S}_T = \sum_{j=1}^{N} (y_j - \bar{y})(y_j - \bar{y})^T = W^T S_T W$$ = Transf. data scatter matrix

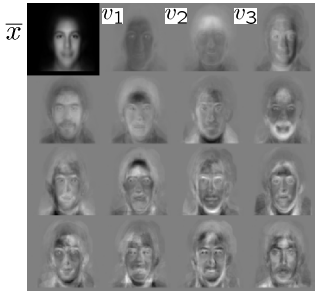
$$W_{opt} = \arg\max_W \left| W^T S_T W \right| = [\overbrace{w_1 \; w_2 \; \ldots \; w_m}^{\text{Eigenvectors of } S_T}]$$



Eigenfaces

- PCA extracts the eigenvectors of $\mathbf{S_T}$
  - Gives a set of vectors $\mathbf{v_1}$, $\mathbf{v_2}$, $\mathbf{v_3}$, ...

  –Each one of these vectors is a direction in face space:

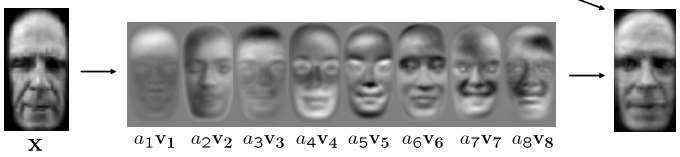


Projecting onto the Eigenfaces

- The eigenfaces $\mathbf{v_1}$, ..., $\mathbf{v_K}$ span the space of faces
  - A face is converted to eigenface coordinates by

$$\mathrm{x} \rightarrow (\underbrace{(\mathrm{x}-\overline{\mathrm{x}})\cdot \mathrm{v}_1}_{a_1}, \underbrace{(\mathrm{x}-\overline{\mathrm{x}})\cdot \mathrm{v}_2}_{a_2}, \ldots, \underbrace{(\mathrm{x}-\overline{\mathrm{x}})\cdot \mathrm{v}_\mathrm{K}}_{a_K})$$

$$\mathrm{x} \approx \overline{\mathrm{x}} + a_1\mathrm{v}_1 + a_2\mathrm{v}_2 + \ldots + a_K\mathrm{v}_\mathrm{K}$$



Algorithm

Training

1. Align training images $x_1, x_2, \ldots, x_N$

Note that each image is formulated into a long vector!

2. Compute average face $u = 1/N \Sigma x_i$

3. Compute the difference image $\varphi_i = x_i - u$

Algorithm

4. Compute the covariance matrix (total scatter matrix)

$$S_T = (1/N) \Sigma \varphi_i \varphi_i^T = BB^T, \quad B=[\varphi_1, \varphi_2 \ldots \varphi_N]$$

5. Compute the eigenvectors of the covariance matrix S_T
6. Compute training projections $a_1, a_2 \ldots a_N$

Testing

1. Take query image X
2. Project X into Eigenface space (W = {eigenfaces}) and compute projection $\omega_i = W(X - u)$,
3. Compare projection ω_i with all training N projections a_i

Illustration of Eigenfaces

- The visualization of eigenvectors:

These are the first 4 eigenvectors from a training set of 400 images (ORL Face Database).

Eigenfaces look somewhat like generic faces.

Reconstruction and Errors

P = 4

P = 200

P = 400

- Only selecting the top P eigenfaces → reduces the dimensionality.
- Fewer eigenfaces result in more information loss, and hence less discrimination between faces.

Summary for Eigenface

Pros

- Non-iterative, globally optimal solution

Limitations

•PCA projection is **optimal for reconstruction** from a low dimensional basis, but **may NOT be optimal for discrimination…**

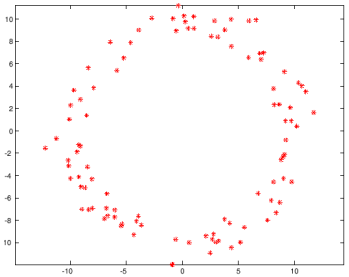
Limitations

- Global appearance method: not robust to misalignment, or background variation



Limitations

- PCA assumes that the data has a Gaussian distribution (mean μ, covariance matrix Σ)



The shape of this dataset is not well described by its principal components

Credit slide: S. Lazebnik

Extensions

•Generalized PCA:

R. Vidal, Y. Ma, and S. Sastry. Generalized Principal Component Analysis (GPCA). IEEE Transactions on Pattern Analysis and Machine Intelligence, volume 27, number 12, pages 1 - 15, 2005.

•Tensor Faces:

"Multilinear Analysis of Image Ensembles: TensorFaces," M.A.O. Vasilescu, D. Terzopoulos, *Proc. 7th European Conference on Computer Vision (ECCV'02)*, Copenhagen, Denmark, May, 2002

•PCA-SIFT

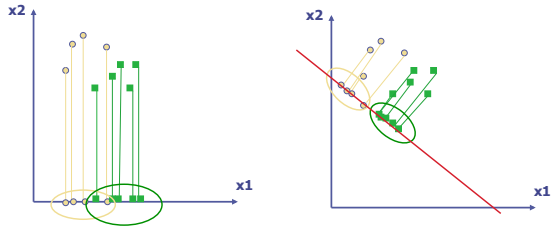
PCA-SIFT: A More Distinctive Representation for Local Image Descriptors - Y Ke, R Sukthankar - IEEE CVPR 04

Linear Discriminant Analysis (LDA)
Fisher's Linear Discriminant (FLD)

- Eigenfaces exploit the max scatter of the training images in face space
- Fisherfaces attempt to maximise the **between class scatter**, while minimising the **within class scatter**.

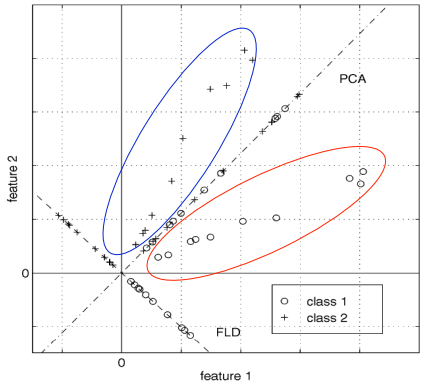
Illustration of the Projection

- Using two classes as example:



Poor Projection Good

Comparing with PCA



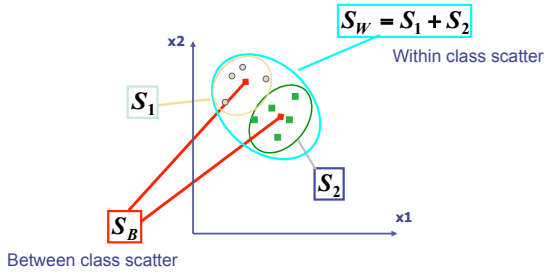
Variables

- N Sample images: $\{x_1, \cdots, x_N\}$
- c classes: $\{\chi_1, \cdots, \chi_c\}$
- Average of each class: $\mu_i = \frac{1}{N_i} \sum_{x_k \in \chi_i} x_k$
- Total average: $\mu = \frac{1}{N} \sum_{k=1}^{N} x_k$

Scatters

- Scatter of class i: $S_i = \sum_{x_k \in \chi_i} (x_k - \mu_i)(x_k - \mu_i)^T$
- Within class scatter: $S_W = \sum_{i=1}^{c} S_i$
- Between class scatter: $S_B = \sum_{i=1}^{c} |\chi_i| (\mu_i - \mu)(\mu_i - \mu)^T$
- Total scatter: $S_T = S_W + S_B$

Illustration



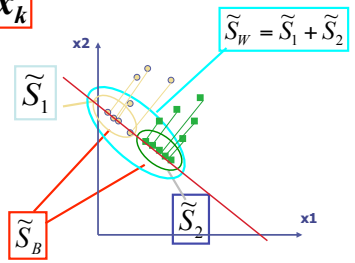
$S_i = \sum_{x_k \in \chi_i} (x_k - \mu_i)(x_k - \mu_i)^T$ $S_W = \sum_{i=1}^{c} S_i$ $S_B = \sum_{i=1}^{c} |\chi_i| (\mu_i - \mu)(\mu_i - \mu)^T$

Mathematical Formulation (1)

- After projection: $y_k = W^T x_k$
- Between class scatter (of y's): $\tilde{S}_B = W^T S_B W$
- Within class scatter (of y's): $\tilde{S}_W = W^T S_W W$

Illustration

$y_k = W^T x_k$



$S_W = \sum_{i=1}^{c} S_i$ $S_B = \sum_{i=1}^{c} |\chi_i| (\mu_i - \mu)(\mu_i - \mu)^T$

$\tilde{S}_W = W^T S_W W$ $\tilde{S}_B = W^T S_B W$

Mathematical Formulation

- The desired projection:

$$W_{opt} = \arg\max_{W} \frac{|\tilde{S}_B|}{|\tilde{S}_W|} = \arg\max_{W} \frac{|W^T S_B W|}{|W^T S_W W|}$$

- How is it found ? → Generalized Eigenvectors
- If S_w has full rank, the generalized eigenvectors are eigenvectors of $S_W^{-1} S_B$ with largest eigen-values

$$S_B w_i = \lambda_i S_W w_i \qquad i = 1, \ldots, m$$

Training/ Testing

Projection in Eigenface

Projection $\omega_i = W_{opt}(X - u)$,

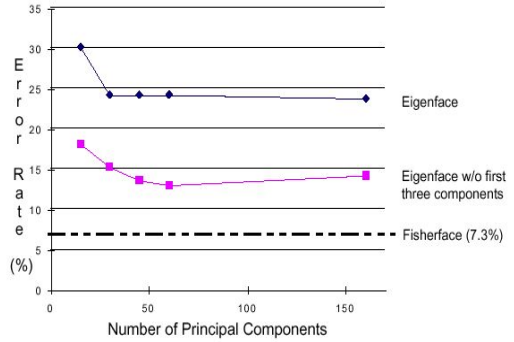
W_{opt} = {fisher-faces}

Results: Eigenface vs. Fisherface (1)

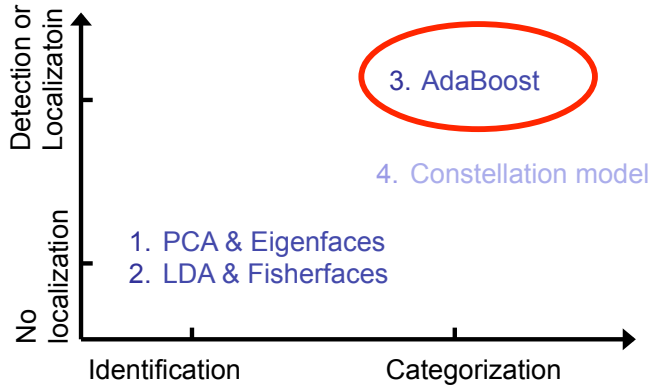
- Input: 160 images of 16 people
- Train: 159 images
- Test: 1 image

- Variation in Facial Expression, Eyewear, and Lighting

Eigenface vs. Fisherface (2)



Face Recognition methods



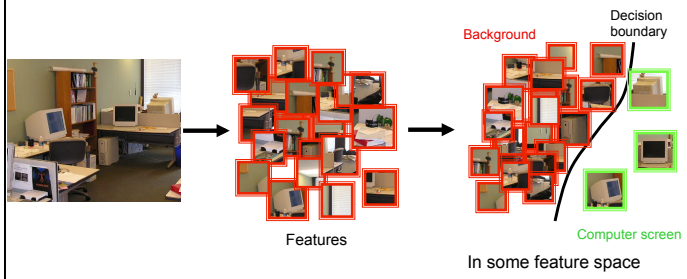
Robust Face Detection Using AdaBoost

- Brief intro on (Ada)Boosting
- Viola & Jones, 2001

Reference:
P. Viola and M. Jones (2001) Robust Real-time Object Detection, IJCV.

Boosting

Designing a strong classifier from a set of weak classifier



Boosting

- Defines a classifier using an additive model:

$$F(x) = \alpha_1 f_1(x) + \alpha_2 f_2(x) + \alpha_3 f_3(x) + \ldots$$

Strong classifier
Features vector
Weight
Weak classifier

Boosting

- Defines a classifier using an additive model:

$$F(x) = \alpha_1 f_1(x) + \alpha_2 f_2(x) + \alpha_3 f_3(x) + \ldots$$

Strong classifier
Features vector
Weight
Weak classifier

- We need to define a family of weak classifiers

$f_k(x)$ form a family of weak classifiers

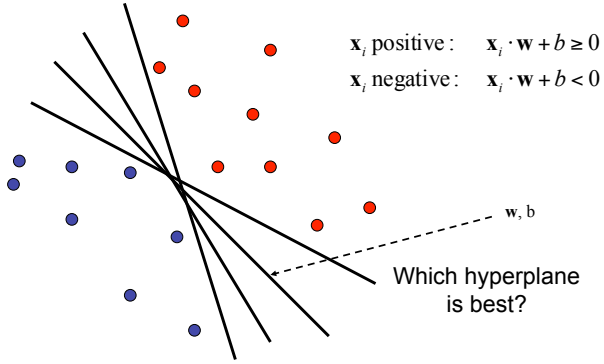
Why boosting?

- A simple algorithm for learning robust classifiers
  - Freund & Shapire, 1995
  - Friedman, Hastie, Tibshhirani, 1998
- Provides efficient algorithm for sparse visual feature selection
  - *Tieu & Viola, 2000*
  - *Viola & Jones, 2003*
- Easy to implement, not requires external optimization tools.

Linear classifiers

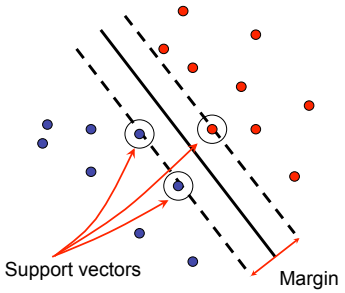
- Find linear function (*hyperplane*) to separate positive and negative examples

$\mathbf{x}_i$ positive: $\mathbf{x}_i \cdot \mathbf{w} + b \geq 0$
$\mathbf{x}_i$ negative: $\mathbf{x}_i \cdot \mathbf{w} + b < 0$



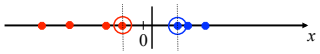
Support vector machines

- Find hyperplane that maximizes the *margin* between the positive and negative examples

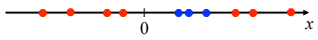


Nonlinear SVMs

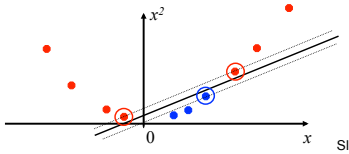
- Datasets that are linearly separable work out great:



- But what if the dataset is just too hard?



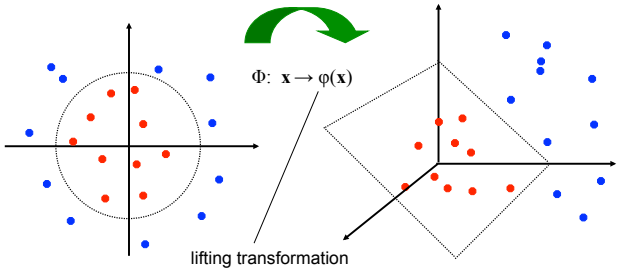
- We can map it to a higher-dimensional space:



Slide credit: Andrew Moore

Nonlinear SVMs

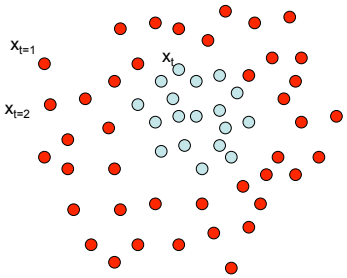
- Map original input space into some higher-dimensional feature space where the training set is separable:



Slide credit: Andrew Moore

Boosting

Y. Freund and R. Schapire, A short introduction to boosting, *Journal of Japanese Society for Artificial Intelligence*, 14(5):771-780, September, 1999.



Each data point has a class label:

$$y_t = \begin{cases} +1 & (\bullet) \\ -1 & (\circ) \end{cases}$$

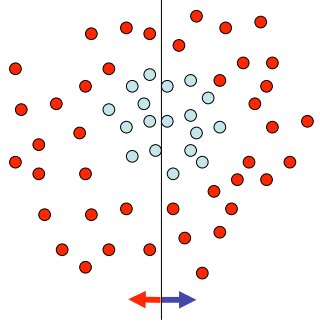
and a weight:

$$w_t = 1$$

- It is a sequential procedure:

Toy example

Weak learners from the family of lines



Each data point has a class label:

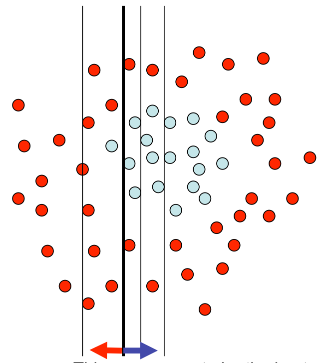
$$y_t = \begin{cases} +1 & (\bullet) \\ -1 & (\circ) \end{cases}$$

and a weight:

$$w_t = 1$$

h => p(error) = 0.5 it is at chance

Toy example



Each data point has a class label:

$y_t = \begin{cases} +1 \; (\bullet) \\ -1 \; (\circ) \end{cases}$

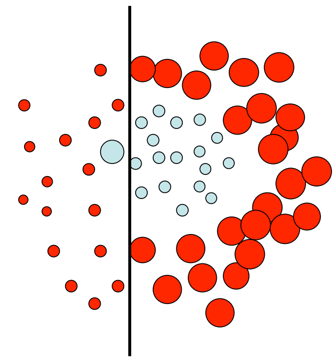
and a weight:

$w_t = 1$

This one seems to be the best

This is a '**weak classifier**': It performs slightly better than chance.

Toy example



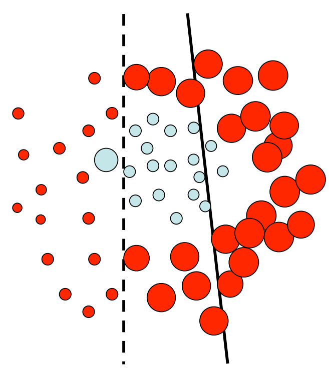
Each data point has a class label:

$y_t = \begin{cases} +1 \; (\bullet) \\ -1 \; (\circ) \end{cases}$

We update the weights:

$w_t \leftarrow w_t \exp\{-y_t H_t\}$

Toy example



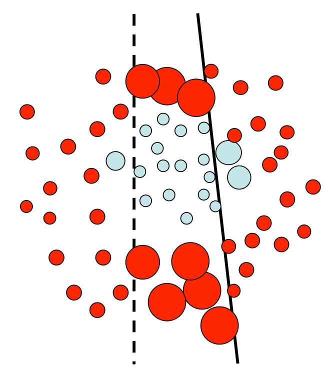
Each data point has a class label:

$y_t = \begin{cases} +1 \; (\bullet) \\ -1 \; (\circ) \end{cases}$

We update the weights:

$w_t \leftarrow w_t \exp\{-y_t H_t\}$

Toy example



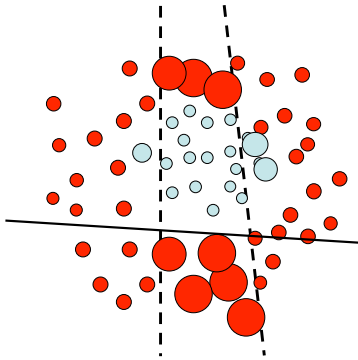
Each data point has a class label:

$y_t = \begin{cases} +1 \; (\bullet) \\ -1 \; (\circ) \end{cases}$

We update the weights:

$w_t \leftarrow w_t \exp\{-y_t H_t\}$

Toy example



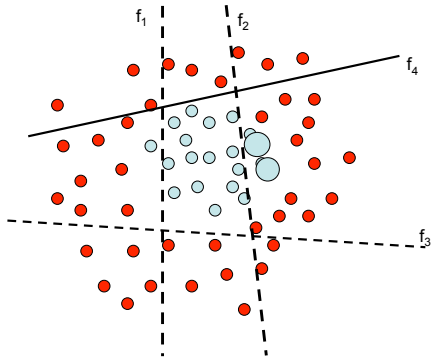
Each data point has a class label:

$y_t = \begin{cases} +1 & (\bullet) \\ -1 & (\circ) \end{cases}$

We update the weights:

$w_t \leftarrow w_t \exp\{-y_t H_t\}$

Toy example



The strong (non- linear) classifier is built as the combination of all the weak (linear) classifiers.

Boosting - mathematics

- Weak learners

value of rectangle feature

$$h_j(x) = \begin{cases} 1 & \text{if } f_j(x) > \theta_j \\ 0 & \text{otherwise} \end{cases}$$

threshold

- Final strong classifier

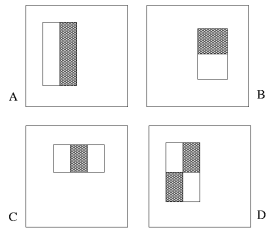
$$h(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 & \text{otherwise} \end{cases}$$

Weak classifier

•4 kind of Rectangle filters

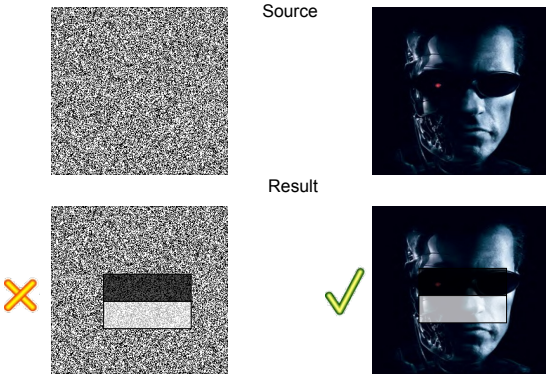
•*Value =*

∑ *(pixels in white area) –*
∑ *(pixels in black area)*



Credit slide: S. Lazebnik

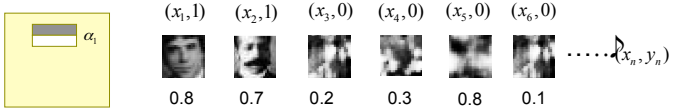
Weak classifier



Credit slide: S. Lazebnik

Weak classifier

1. Evaluate each rectangle filter on each example



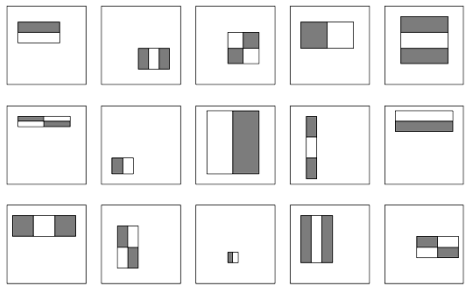
$$h_j(x) = \begin{cases} 1 & \text{if } f_j(x) > \theta_j \\ 0 & \text{otherwise} \end{cases}$$

threshold

Weak classifier

Weak classifier

- For a 24x24 detection region,



Boosting for face detection

2. Select best filter/threshold combination

a. Normalize the weights $w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}}$

$h_j(x) = \begin{cases} 1 & \text{if } f_j(x) > \theta_j \\ 0 & \text{otherwise} \end{cases}$

b. For each feature, j $\varepsilon_j = \sum_i w_i \left| h_j(x_i) - y_i \right|$

c. Choose the classifier, h_t with the lowest error ε_t

3. Reweight examples

$$w_{t+1,i} = w_{t,i} \beta_t^{1-|h_t(x_i)-y_i|} \qquad \beta_t = \frac{\varepsilon_t}{1-\varepsilon_t}$$

Boosting for face detection

4. The final strong classifier is

$$h(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 & \text{otherwise} \end{cases} \qquad \alpha_t = \log \frac{1}{\beta_t}$$

The final hypothesis is a weighted linear combination of the T hypotheses where the weights are inversely proportional to the training errors

Feature selection

Non discriminative features

Discriminative features

The Viola/Jones Face Detector

P. Viola and M. Jones. *Rapid object detection using a boosted cascade of simple features.* CVPR 2001.

- A "paradigmatic" method for real-time object detection
- Training is slow, but detection is very fast
- Key ideas
  - *Integral images* for fast feature evaluation
  - *Boosting* for feature selection
  - *Attentional cascade* for fast rejection of non-face windows

Credit slide: S. Lazebnik

Boosting for face detection

- For each round of boosting:
  1. Evaluate each rectangle filter on each example
  2. Select best filter/threshold combination
  3. Reweight examples

The implemented system

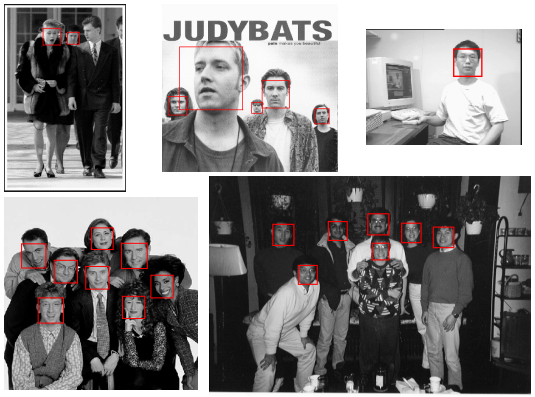
- Training Data
  - 5000 faces
    - All frontal, rescaled to 24x24 pixels
  - 300 million non-faces
    - 9500 non-face images
  - Faces are normalized
    - Scale, translation
- Many variations
  - Across individuals
  - Illumination
  - Pose

(Most slides from Paul Viola)

System performance

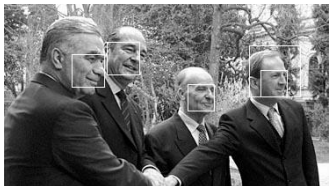
- Training time: “weeks” on 466 MHz Sun workstation
- 38 layers, total of 6061 features
- Average of 10 features evaluated per window on test set
- “On a 700 Mhz Pentium III processor, the face detector can process a 384 by 288 pixel image in about .067 seconds”
  - 15 Hz
  - 15 times faster than previous detector of comparable accuracy (Rowley et al., 1998)

Output of Face Detector on Test Images



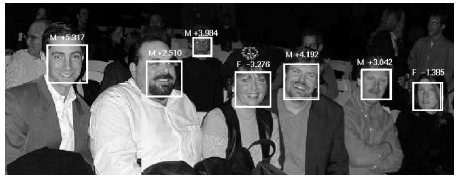
Other detection tasks

Facial Feature Localization

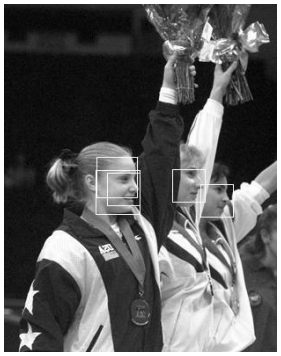


Profile Detection

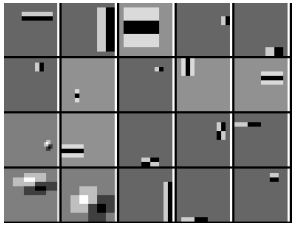
Male vs. female



Profile Detection



Profile Features



Face Image Databases

- Databases for face recognition can be best utilized as training sets
  - Each image consists of an individual on a uniform and uncluttered background
- Test Sets for face detection
  - MIT, CMU (frontal, profile), Kodak

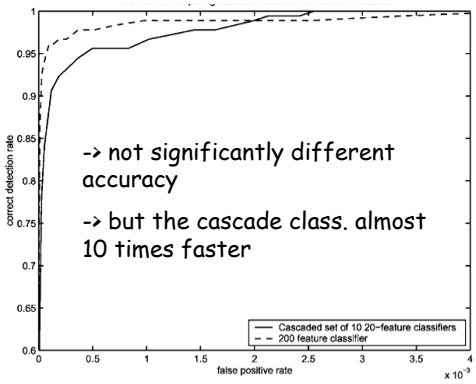
Experimental Results

- Test dataset
  - MIT+CMU frontal face test set
  - 130 images with 507 labeled frontal faces

False detection	10	31	50	65	78	95	110	167	422
AdaBoost	78.3	85.2	88.8	89.8	90.1	90.8	91.1	91.8	93.7
Neural-net	83.2	86.0	-	-	-	89.2	-	90.1	89.9

MIT test set: 23 images with 149 faces♪
Sung & poggio: detection rate 79.9% with 5 false positive♪
AdaBoost: detection rate 77.8% with 5 false positives ♪
♪

Experimental Results

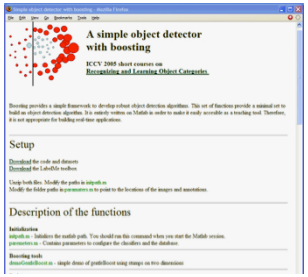


-> not significantly different accuracy

-> but the cascade class. almost 10 times faster

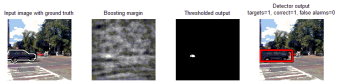
Sharing features with Boosting

Sharing features: efficient boosting procedures for multiclass object detection
A. Torralba, K. P. Murphy and W. T. Freeman Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR). pp 762-769, 2004.



Matlab code

- Gentle boosting
- Object detector using a part based model



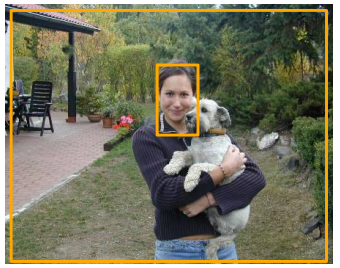
http://people.csail.mit.edu/torralba/iccv2005/

EECS 442 – Computer vision

Next lecture: scene understanding

Recognition

– Recognition task: classification, detection, etc..



Recognition

– Recognition task

– Search strategy: Sliding Windows Viola, Jones 2001,

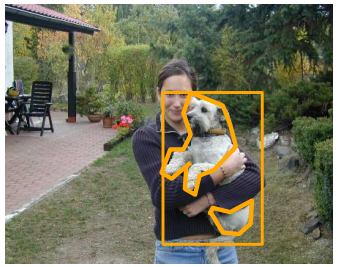
- Simple
- Computational complexity
  - BSW by Lampert et al 08
  - Also, Alexe, et al 10

Recognition

– Recognition task

– Search strategy: Sliding Windows Viola, Jones 2001,

- Simple
- Computational complexity
  - BSW by Lampert et al 08
  - Also, Alexe, et al 10
- Localization
  - Objects are not boxes



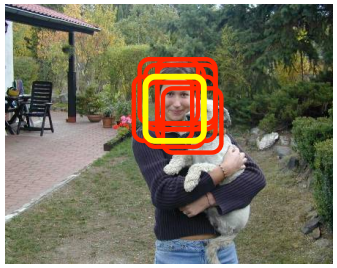
Recognition

– Recognition task

– Search strategy: Sliding Windows Viola, Jones 2001,

- Simple
- Computational complexity
  - BSW by Lampert et al 08
  - Also, Alexe, et al 10
- Localization
  - Objects are not boxes
  - Prone to false positive

Non max suppression:
Canny '86
….
Desai et al , 2009

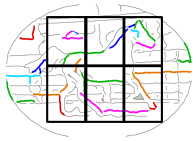


Successful methods using sliding windows

- Subdivide scanning window
- In each cell compute histogram of gradients orientation.

Code available: http://pascal.inrialpes.fr/soft/olt/

[Dalal & Triggs, CVPR 2005]

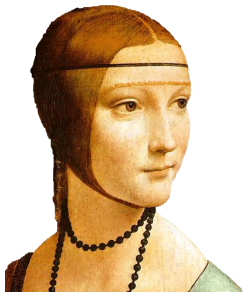


- Subdivide scanning window
- In each cell compute histogram of codewords of adjacent segments

Code available: http://www.vision.ee.ethz.ch/~calvin

[Ferrari & al, PAMI 2008]

Object → **Bag of 'words'**



Analogy to documents

Of all the sensory impressions proceeding to the brain, the visual experiences are the dominant ones. Our perception of the world around us is based essentially on the messages that r… our eyes. For a long ti… etinal image wa… isual centers … s a movie … image … discov… know t… percepti… more com… following the … ath to the various … ortex, Hubel and Wiesel … demonstrate that the *message abo… image falling on the retina undergoes …-wise analysis in a system of nerve cel… stored in columns. In this system each … has its specific function and is responsibl… a specific detail in the pattern of the retinal image.*

sensory, brain, visual, perception, retinal, cerebral cortex, eye, cell, optical nerve, image Hubel, Wiesel

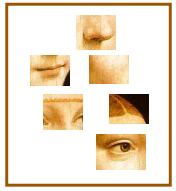
China is forecasting a trade surplus of $90bn (£51bn) to $100bn this year, a threefold increase on 2004's $32bn. The Commerce Ministry said the surplus would be created by a predicted 30% … $750bn, compared … $660bn. T… annoy t… China's … deliber… agrees … yuan is … governo… also need… demand so … country. China … e yuan against the d… nd permitted it to trade within a narrow … but the US wants the yuan to be allowed … de freely. However, Beijing has made it c… t it will take its time and tread carefully be… allowing the yuan to rise further in value.

China, trade, surplus, commerce, exports, imports, US, yuan, bank, domestic, foreign, increase, trade, value

definition of "BoW"

– Independent features

face



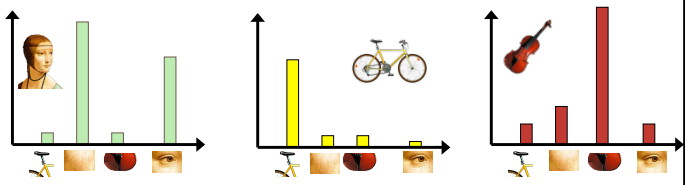
bike

violin

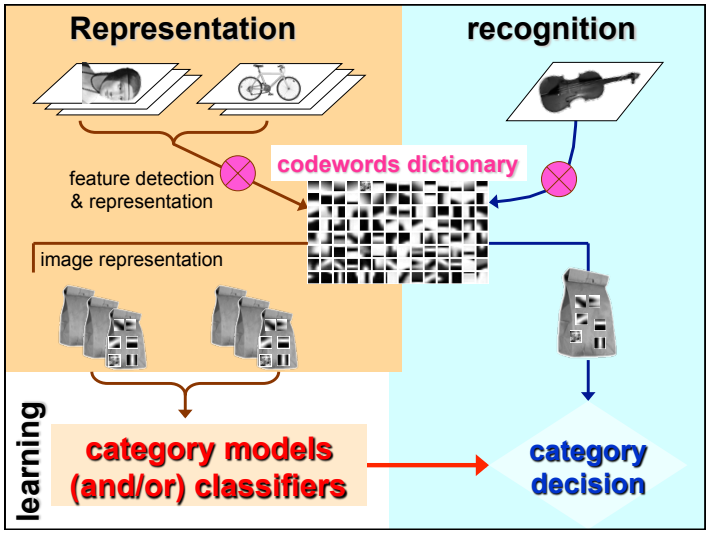
definition of "BoW"

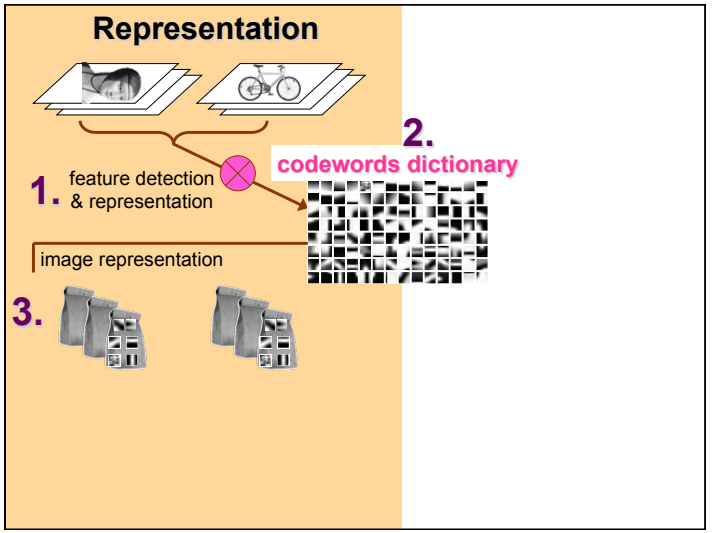
– Independent features

– histogram representation

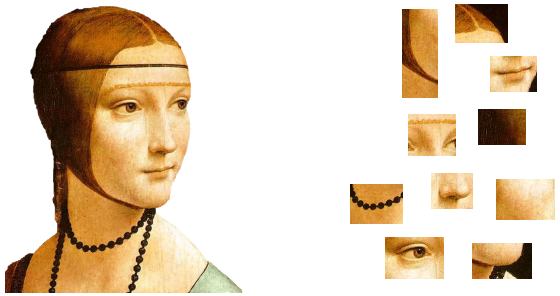


codewords dictionary





1.Feature detection and description

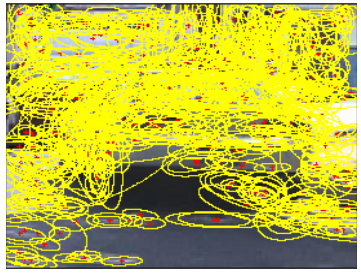


1.Feature detection and description

- Regular grid
  - Vogel & Schiele, 2003
  - Fei-Fei & Perona, 2005

1.Feature detection and description

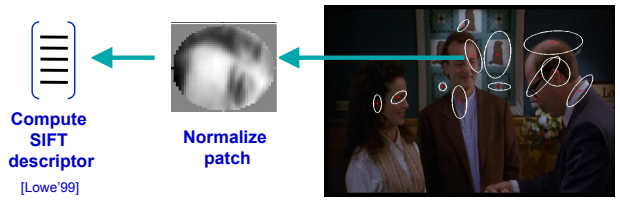
- Regular grid
  - Vogel & Schiele, 2003
  - Fei-Fei & Perona, 2005
- Interest point detector
  - Csurka, et al. 2004
  - Fei-Fei & Perona, 2005
  - Sivic, et al. 2005



1.Feature detection and description

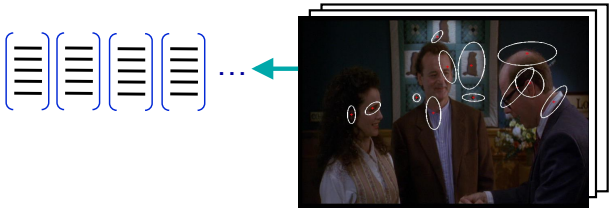
- Regular grid
  - Vogel & Schiele, 2003
  - Fei-Fei & Perona, 2005
- Interest point detector
  - Csurka, Bray, Dance & Fan, 2004
  - Fei-Fei & Perona, 2005
  - Sivic, Russell, Efros, Freeman & Zisserman, 2005
- Other methods
  - Random sampling (Vidal-Naquet & Ullman, 2002)
  - Segmentation based patches (Barnard, Duygulu, Forsyth, de Freitas, Blei, Jordan, 2003)

1.Feature detection and description

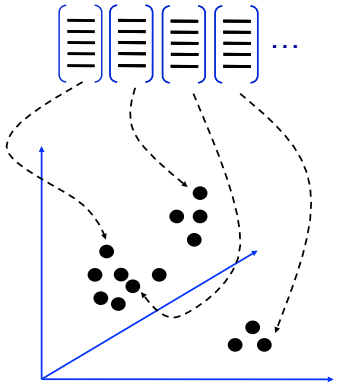


Slide credit: Josef Sivic

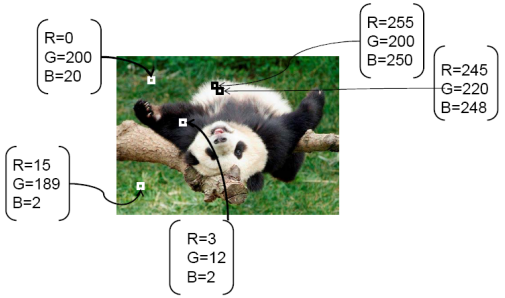
2. Codewords dictionary formation



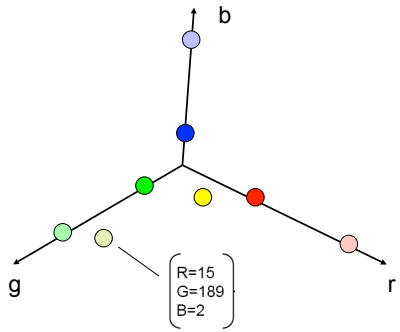
2. Codewords dictionary formation



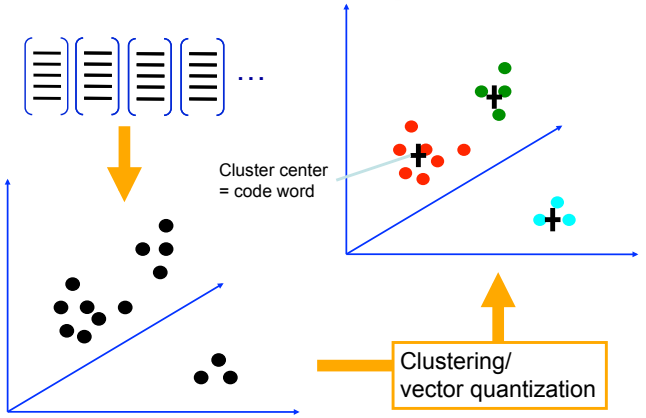
Example: color feature



Example: color feature

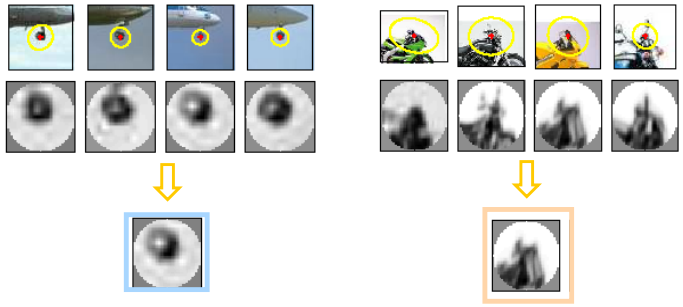


2. Codewords dictionary formation



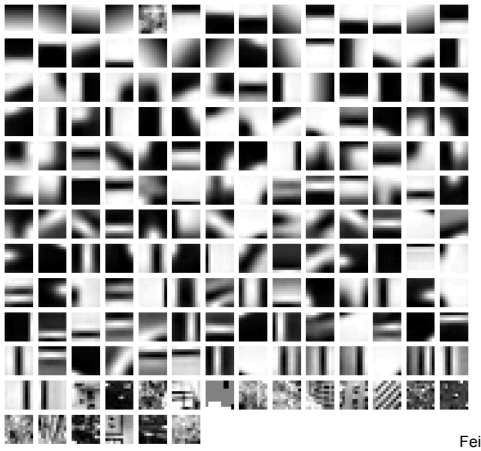
2. Codewords dictionary formation

- **Image patch examples of codewords**



Sivic et al. 2005

2. Codewords dictionary formation



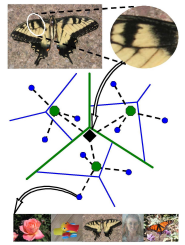
Fei-Fei et al. 2005

2. Codewords dictionary formation

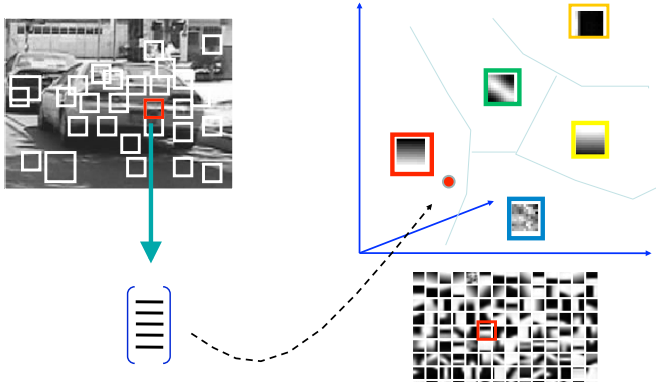
- **Typically a codeword dictionary is obtained from a training set comprising all the object classes of interests**

Visual vocabularies: Issues

- How to choose vocabulary size?
  - Too small: visual words not representative of all patches
  - Too large: quantization artifacts, overfitting
- Computational efficiency
  - Vocabulary trees (Nister & Stewenius, 2006)



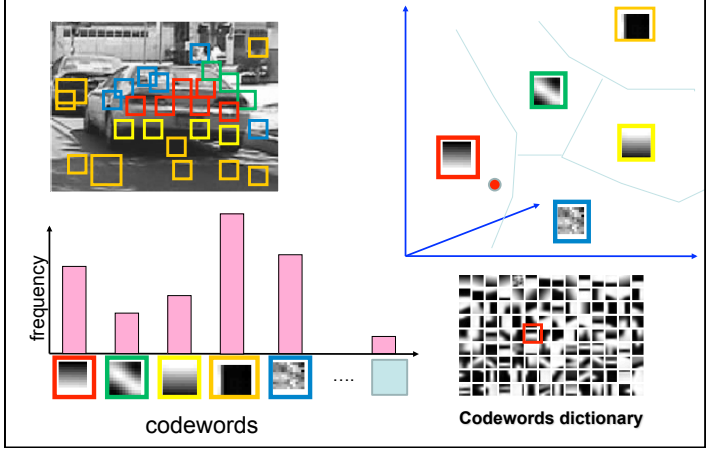
3. Bag of word representation



- Nearest neighbors assignment
- K-D tree search strategy

Codewords dictionary

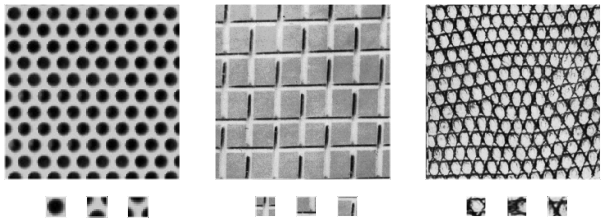
3. Bag of word representation



Codewords dictionary

Representing textures

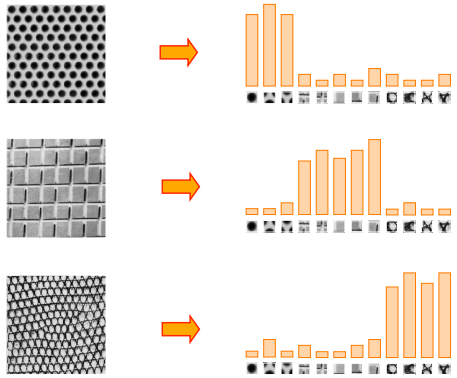
- Texture is characterized by the repetition of basic elements or *textons*
- For stochastic textures, it is the identity of the textons, not their spatial arrangement, that matters



Julesz, 1981; Cula & Dana, 2001; Leung & Malik 2001; Mori, Belongie & Malik, 2001; Schmid 2001; Varma & Zisserman, 2002, 2003; Lazebnik, Schmid & Ponce, 2003

Credit slide: S. Lazebnik

Representing textures



Julesz, 1981; Cula & Dana, 2001; Leung & Malik 2001; Mori, Belongie & Malik, 2001; Schmid 2001; Varma & Zisserman, 2002, 2003; Lazebnik, Schmid & Ponce, 2003

Credit slide: S. Lazebnik

Invariance issues

- Scale? Rotation? View point? Occlusions?
  - Implicit;
  - depends on detectors and descriptors

Kadir and Brady. 2003

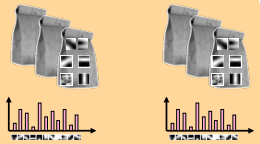
Representation

2. codewords dictionary

1. feature detection & representation

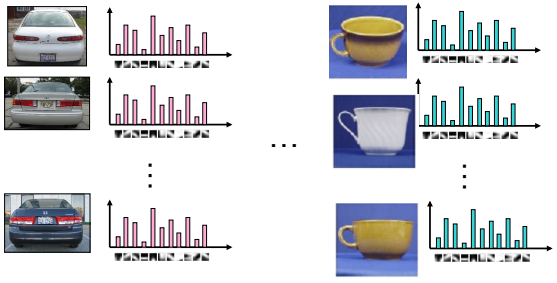
image representation

3.



category models

Category models



Class 1

Class N

Recognition

codewords dictionary

category models (and/or) classifiers

category decision

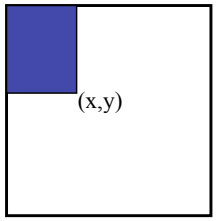
Feature selection

- For a 24x24 detection region, the number of possible rectangle features is ~180,000!
- At test time, it is impractical to evaluate the entire feature set
- Can we create a good classifier using just a small subset of all possible features?
- How to select such a subset?

Credit slide: S. Lazebnik

Fast computation with integral images

- The *integral image* computes a value at each pixel (*x*,*y*) that is the sum of the pixel values above and to the left of (*x*,*y*), inclusive
- This can quickly be computed in one pass through the image



$$ii(x,y) = \sum_{x' \le x, y' \le y} i(x',y')$$

Credit slide: S. Lazebnik

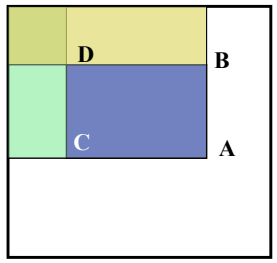
Fast computation with integral images

Computing sum within a rectangle

- Let A,B,C,D be the values of the integral image at the corners of a rectangle

•Then the sum of original image values within the rectangle can be computed as:

sum = A – B – C + D



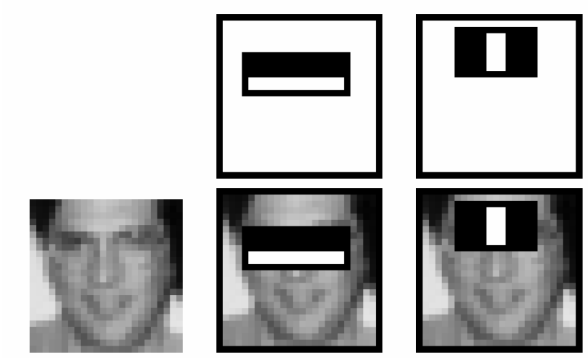
•Only 3 additions are required for any size of rectangle!

–This is now used in many areas of computer vision

–E.g. Histogram of features in rectangular areas

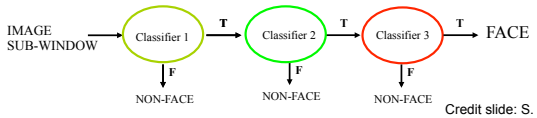
Credit slide: S. Lazebnik

First two features selected by boosting



Cascading classifiers

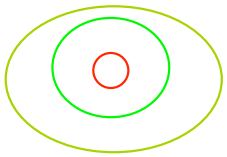
- We start with simple classifiers which reject many of the negative sub-windows while detecting almost all positive sub-windows
- Positive results from the first classifier triggers the evaluation of a second (more complex) classifier, and so on
- A negative outcome at any point leads to the immediate rejection of the sub-window



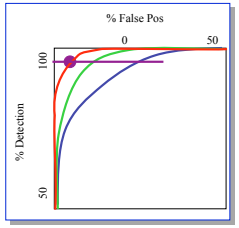
Credit slide: S. Lazebnik

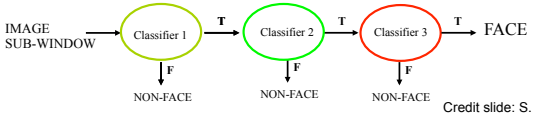
Cascading classifiers

- Chain classifiers that are progressively more complex and have lower false positive rates:



Receiver operating characteristic





Credit slide: S. Lazebnik

Training the cascade

- Adjust weak learner threshold to minimize *false negatives* (as opposed to total classification error)
- Each classifier trained on false positives of previous stages
  - A single-feature classifier achieves 100% detection rate and about 50% false positive rate
  - A five-feature classifier achieves 100% detection rate and 40% false positive rate (20% cumulative)
  - A 20-feature classifier achieve 100% detection rate with 10% false positive rate (2% cumulative)

Credit slide: S. Lazebnik

Summary: Viola/Jones detector

- Rectangle features
- Integral images for fast computation
- Boosting for feature selection
- Attentional cascade for fast rejection of negative windows